

## IN THE CLAIMS:

**Please amend claim 1 and add new Claims 2-20 as follows.**

1. (Currently Amended) A computer-readable storage medium, the computer readable storage medium having stored thereon instructions executable to cause one or more processors to: debugger for use in assisting in the debugging of a program in a computer system, the debugger comprising:

A. a thread identifier configured to, execute, on a first processor, two or more threads associated with a program,

in response to a breakpoint event in connection with a breakpoint in a first thread which has caused an operating system to stop execution of [[all]] the two or more threads associated with the [[in the]] program, identifies

identify, by a debugger, the first thread as containing the breakpoint, one of said threads which contained the breakpoint, and B. a thread execution controller configured to, after identifying the one of said threads

transfer control of the two or more threads associated with the program from the operating system to the debugger,

enable others of [[said]] the two or more threads to continue execution resume executing by passing control of the others of the two or more threads back to the operating system,

wherein the others of the two or more threads do not contain the breakpoint instruction, and simultaneously execute the first thread under control of the debugger, wherein the debugger controls subsequent processing operations in connection with the first thread by receiving commands from a user, and

wherein the commands direct the debugger to step through the first thread.

2. (New) The computer-readable storage medium of claim 1, further comprising instructions executable to cause the one or more processors to:  
initialize an execution environment, the operating system, and the debugger.
3. (New) The computer-readable storage medium of claim 2, further comprising instructions executable to cause the one or more processors to:  
initialize the program; and  
in response to a first request from the program, create, under control of the debugger, a  
second thread among the one or more threads.
4. (New) The computer-readable storage medium of claim 3, further comprising instructions executable to cause the one or more processors to:  
in response to a second request from the program, delete, under the control of the  
debugger, the second thread.
5. (New) The computer-readable storage medium of claim 3, wherein the first thread is  
executed in a first address space and the second thread is executed in a second address  
space.

6. (New) The computer-readable storage medium of claim 3, wherein the first thread is executed by the first processor and the second thread is executed by a second processor among the one or more processors.
7. (New) The computer-readable storage medium of claim 6, further comprising instructions executable to cause the one or more processors to:  
display information generated by the first thread and the second thread.
8. (New) A method, comprising:  
executing, on a first processor, two or more threads associated with a program;  
responsive to the first processor executing a breakpoint instruction, stopping execution of  
the two or more threads associated with the program;  
transferring control of the two or more threads associated with the program from an  
operating system to a debugger;  
identifying as containing the breakpoint instruction, by the debugger, a first thread from  
among the two or more threads associated with the program;  
resuming execution of other threads from among the two or more threads associated with  
the program by passing control of the other threads back to the operating system,  
wherein the other threads do not contain the breakpoint instruction; and  
simultaneously executing the first thread under control of the debugger, wherein the  
debugger controls subsequent processing operations in connection with the first  
thread by receiving commands from a user, and  
wherein the commands direct the debugger to step through the first thread.

9. (New) The method of claim 8, further comprising:  
initializing an execution environment, the operating system, and the debugger.
10. (New) The method of claim 9, further comprising:  
initializing the program; and  
in response to a first request from the program, the debugger creating a second thread  
under control of the debugger among the two or more threads.
11. (New) The method of claim 10, further comprising:  
in response to a second request from the program, the debugger deleting the second  
thread.
12. (New) The method of claim 11, wherein the first thread is executed in a first address  
space and the second thread is executed in a second address space.
13. (New) The method of claim 11, wherein the first thread is executed by the first processor  
and the second thread is executed by a second processor.
14. (New) The method of Claim 13, further comprising displaying information generated by  
the first thread and the second thread.

15. (New) A system, the system comprising:

one or more processors;

a memory storing program instructions, wherein the program instructions are executable to cause

the one or more processors to execute a method comprising:

executing, on a first processor among the one or more processors, two or more threads

associated with a program;

responsive to the first processor executing a breakpoint instruction, stopping execution of

the two or more threads associated with the program;

transferring control of the two or more threads associated with the program from an

operating system to a debugger;

identifying as containing the breakpoint instruction, by the debugger, a first thread from

among the two or more threads associated with the program;

resuming execution of other threads from among the two or more threads associated with

the program by passing control of the other threads back to the operating system,

wherein the other threads do not contain the breakpoint instruction; and

simultaneously executing the first thread under control of the debugger, wherein the

debugger controls subsequent processing operations in connection with the first

thread by receiving commands from a user, and

wherein the commands direct the debugger to step through the first thread.

16. (New) The system of claim 15, wherein the method further comprises

initializing an execution environment, the operating system, and the debugger.

17. (New) The system of claim 16, wherein the method further comprises  
initializing the program; and  
in response to a first request from the program, the debugger creating a second thread  
under control of the debugger among the two or more threads.
18. (New) The system of claim 17, wherein the method further comprises  
in response to a second request from the program, the debugger deleting the second  
thread.
19. (New) The system of claim 18, wherein the first thread is executed in a first address  
space and the second thread is executed in a second address space.
20. (New) The system of claim 18, wherein the first thread is executed by the first processor  
and the second thread is executed by a second processor.